www.aspectsecurity.com

ASPECT SECURITY
Application Security Experts

GSA Schedule Contract Number: GS-35F-0082U

SIN 132-50 Instructor Lead Training

Building and Testing Secure Web Applications
Course Outline
(pages 2-11)

SIN 132-51 Professional Services

Application Security Verification Service
(pages 12-18)

SIN 132-32 Term Software Licenses

Contrast – Application Security Verification Tool
(pages 19-20)

SIN 132-32 Term Software Licenses

eLearning Modules for Application Security Training
(pages 21-33)

Pricing – Pages 34-35

Aspect Security, Inc.

Contact: Bill Husted
Email: bill.husted@aspectsecurity.com
Phone: (direct) 301-775-5545
(office) 301-604-4882

9175 Guilford Road, Suite 300
Columbia, MD 21046
office 301.604.4882
fax 443.583.0772

1 Training Executive Summary

Web application vulnerabilities continue to place our global computing infrastructure at risk. In this course, students will perform hands-on security testing on live web applications to find common vulnerabilities and will learn efficient and effective approaches for eliminating or avoiding these vulnerabilities in your web applications. Students will learn how to diagnose all of the OWASP Top Ten web flaws, including Cross-Site Scripting (XSS), SQL Injection, Cross-Site Request Forgery, Broken Authentication and Authorization, and much more.

The course is designed primarily for software developers and testers, but anyone with an interest in web application security will be able to use the tools provided and learn to find and diagnose holes in a real web application. Each student receives a CD with an application security learning environment and a number of specialized tools. The course culminates with a fun three-stage challenge designed to drive home the key lessons from the training.

This course goes way beyond just finding and exploiting vulnerabilities. Students will also learn about the security controls that developers can use to solve these issues. Understanding how security is supposed to work is the greatest tool you can possibly have for finding security problems. The instructors are experienced application security professionals who come to class ready to share their real-world experiences and decades of expertise. We make the courses relevant to your organization by discussing how to apply the course's lessons learned to your current challenges.

2 Audience

- Software Developers (any web environment)
- Software Testers
- Security Specialists
- Application Architects

3 Learning Objectives

At the highest level, the objective for this course is to ensure that developers are capable of designing, building, and testing secure applications and understand why this is important.

Topic	Learning Objective
HTTP Fundamentals	Understand and be able to employ the security features involved with using HTTP (e.g., headers, cookies, SSL)
Design Principles and Patterns	Understand and be able to apply application security design principles.
Threats	Be able to identify and explain common web application security threats (e.g., Cross-Site Scripting, SQL Injection, Access Control Attacks, "Man-in-the-middle" attacks, etc.) and implement mitigation techniques.
Authentication and Session Management	Be able to handle credentials securely while providing the full range of authentication support functions, including login, change password, forgot password, remember password, logout, re-authentication, and timeouts.

Access Control	Be able to implement access control rules for the user interface, business logic, and data layers.
Cross-Site Request Forgery (CSRF)	Learn how Cross-Site Request Forgery attacks work, the serious damage they can cause, and easy defenses against this type of attack.
Cross-Site Scripting (XSS)	Be able to implement simple, straightforward Cross-Site Scripting defenses through proper output encoding and how to detect XSS vulnerabilities throughout an application.
Clickjacking	Learn how Clickjacking attacks work, and the security headers that can now be used to easily defend against them.
Input Validation	Learn the strong benefits of proper input validation, and how to architect an effective and easy to use input validation framework.
Command Injection	Understand the dangers of command injection and techniques for avoiding the introduction of this type of vulnerability.
Error Handling	Be able to implement a consistent error (exception) handling and logging approach for an entire web application.
Cryptography	Learn when to apply cryptographic techniques and be able to choose algorithms and use encryption/decryption and hash functions securely.
Auditing and Logging	Be able to select and implement appropriate auditing/logging capabilities.
Avoid Using Vulnerable Components	The OWASP Top 10 for 2013 raised awareness to the world that a very common source of vulnerabilities in web applications is the use of known vulnerable components. The course describes how to avoid this issue with the use of automation.
Verification	Be able to review their applications for common security vulnerabilities using code review and penetration testing techniques.

4 About the Instructors

Aspect Security has been working with development teams around the world for over a decade to help them identify, diagnose, and address security issues throughout the application development lifecycle. Through these efforts, they have learned the key practices that development and project managers, and key support personnel must know to achieve secure applications.

Aspect's instructors are full-time application security specialists who spend the majority of their time working with clients to secure the nation's most critical applications. Leveraging this practical experience brings the class to life. Students will gain valuable insight into lessons learned from other development organizations. Our instructors are available to you to answer your application security questions after the course is complete.

Aspect is a Founding OWASP Member and supports numerous OWASP projects. In particular, Aspect conceived of the OWASP Top Ten project in 2003 and still leads this project. Aspect staff also created WebGoat, Stinger, AntiSamy, and CSRF Guard and donated them to the OWASP effort. Aspect's founders established the OWASP Foundation in 2004, launched the OWASP AppSec conference series in 2005, and continue to make significant contributions to OWASP. One of our largest OWASP contributions is the Enterprise Security API (ESAPI) project, which is the definition of all the security functions that developers need on a daily basis to build secure web applications, and then a reference implementation of these functions.

In 2009, we launched the OWASP Cheat Sheet series, which provides simple/straightforward advice on how to implement numerous security controls.

5 Agenda

The exact agenda can vary depending upon the class and the organization's objectives. The instructor will adjust the course's emphasis based on the skills and interest of the class, as well as the technologies and environments used in the company.

Day 1	
8:30	Welcome and Introductions
8:40	Understanding HTTP ▲
9:00	
9:20	
9:40	How to Authenticate Users
10:00	BREAK
10:20	How to Authenticate Users (cont) ▲
10:40	How to Manage User Sessions ▲
11:00	
11:20	How to Control Access ▲
11:40	
12:00	LUNCH
12:20	
12:40	
13:00	How to Control Access (cont) ▲
13:20	How to Protect Against CSRF ▲
13:40	
14:00	How to Protect Against XSS
14:20	BREAK
14:40	How to Protect Against XSS (cont) ▲
15:00	
15:20	
15:40	How to Defend Against Clickjacking ▲
16:00	
16:20	Wrap-Up Day 1
16:30	

Day 2	
8:30	Welcome to Day 2
8:40	How to Architect Input Validation ▲
9:00	
9:20	How to Protect Sensitive Data ▲
9:40	How to Use Databases Securely
10:00	BREAK
10:20	How to Use Databases Securely (cont.) ▲
10:40	
11:00	
11:20	Error Handling and Logging ▲
11:40	
12:00	LUNCH
12:20	
12:40	
13:00	Components with Known Vulnerabilities
13:20	How to Access Services Securely ▲
13:40	
14:00	HTML5 Security ▲
14:20	
14:40	BREAK
15:00	AppSec at DevOps Speed and Portfolio Scale
15:20	
15:40	Challenge ▲
16:00	
16:20	Wrap-Up Day 2
16:30	

▲ Hands-on Testing Exercise

6 Outline

1) Introduction

Section Overview: This section describes and introduces the course and instructors. It also provides setup instructions for the course exercises.

a) Training Program Introduction
b) Course Objectives, Approach, and Layout
c) Intro to Aspect Security/Instructors
d) Students Introduce Themselves
e) Discussion of Applicable Corporate Initiatives
f) Review of Course Agenda
g) Install and Setup Testing Environment

2) Understanding HTTP and Web Technologies

Section Overview: This section is intended to provide the foundations needed to understand the upcoming application security concepts. It begins by describing the HTTP protocol and how it relates to web applications. It dives into various aspects of the protocol, in detail, to assist in the understanding of the entire communication path from client request, server processing, server response, and browser interpretation. It then discusses how a hacker proxy can be used to modify HTTP requests and where this proxy fits into the big picture. Finally, we begin the first hands-on lesson, which is intended to get the students familiar with the hands-on application and comfortable using the testing proxy.

a) HTTP Protocol (Requests, Responses, Headers, Cookies, Parameters, Response Codes)
    i) Security of GET vs. POST
    ii) SSL and Certificates
    iii) Man-in-the-Middle Threat
    iv) HTTP Strict Transport Security
b) Introducing Test Application and Security Testing Proxy
    i) WebGoat Overview
    ii) ZAP Overview
c) Exercise and Labs
    i) Hands-On Testing Exercise: WebGoat HTTP Basics
    ii) Hands-On Testing Exercise: WebGoat and Proxy

3) How to Authenticate Users

Section Overview: This section includes common web authentication methods along with their strengths and weaknesses. It discusses best practices associated with authentication and uses hands-on lessons to demonstrate common authentication mistakes. Through this we discuss different technology specific authentication uses and configuration.

a) Overview
b) Authentication Mechanisms
c) Common Authentication Approaches
d) How to Protect Credentials from Disclosure
e) How to Protect Against Brute Force Attacks
f) How to Provide Password Management Functions
g) Exercises and Labs
    i) Hands-On Testing Exercise: WebGoat – Basic Authentication

4) How to Manage User Sessions

Section Overview: This section includes what session management is and how it works within a web application environment. It discusses common mistakes developers make regarding session management and attacks that can take advantage of these errors. The section discusses best practices associated with session management and technology specific implementation approaches.

a) Introduction to HTTP Sessions
b) Explanation of Session Lifecycle (login, logout, reauthentication, timeouts)
c) How to Protect Against Session Hijacking
d) Exercise and Labs
    i) Hands-On Testing Exercise: WebGoat – Authentication Cookies

5) How to Control Access

Section Overview: This section introduces Access Control in a web environment and the various complexities associated with implementing strong access protections. It walks through the importance of checking all access to sensitive functionality, defining application roles and functions, not relying only on presentation rendering, and implementing access controls at different level, including: declarative (URL), programmatic (API) and instance (data) level. Throughout the section, various technology specific access control uses are discussed and demonstrated. This section also includes common best practices associates with access control.

a) Overview
b) Defining & Architecting Your Access Control Policy
    i) Authorization Primitives
    ii) Defining an Access Control matrix
c) Presentation Layer Access Control
    i) Single Role vs. Multi-Role Views
d) Environment Enforced Access Control
    i) Attack Surface
    ii) Single Role vs. multi-Role URLs
    iii) Declarative Authorization
e) Business Layer Access Control
    i) Programmatic Authorization
    ii) Single Role vs. Multi-Role Business Functions
f) Data Layer Access Control
    i) The Object Reference Problem
g) Exercise and Labs
    i) Hands-On Testing Exercise: WebGoat – Access Control

6) How to Prevent Cross-Site Request Forgery (CSRF) Attacks

Section Overview: The section introduces a very common web application attack most developers aren't familiar with known as Cross-Site Request Forgery. It explains how and why this attack works and the consequences of such attacks. It discusses the significance of these types of flaws and presents several approaches for how developers can defend their applications against this type of attack.

a) Overview of CSRF
    i) What is CSRF
    ii) CSRF Vulnerability Pattern
    iii) The Same Origin Policy

b) How to Identify CSRF Flaws
   i) Several Real World Examples
c) How to Protect Against CSRF
   i) Misconceptions – Defenses That Don't Work
   ii) Recommended CSRF Defenses
   iii) Protecting REST Interfaces Against CSRF
   iv) Java and .Net Specific Defenses

7) How to Protect Against Cross-Site Scripting

Section Overview: The section covers in detail a very common web application attack known as Cross-Site Scripting (XSS). It explains how and why this attack works and the consequences of such attacks. It introduces and explains two types of XSS attacks (reflected and stored), demonstrates an attack, walks through various buggy code examples. And finally allow the students to apply what they have learned by executing XSS attacks using hands-on lessons. Throughout the section different technology specific protections, including output encoding and input validation, are explored and discussed.

a) Overview of XSS
   i) Types of XSS (Stored and Reflected)
   ii) Tricking the Browser Sandbox
   iii) Consequences of XSS
b) How to Solve XSS Problems
   i) Output Encoding
   ii) Input Validation
   iii) Filters
   iv) HTTP Only
   v) Response Headers to Help Prevent XSS
c) Exercises and Labs
   i) Hands-On Testing Exercise: WebGoat – Stored and Reflected XSS
   ii) Hands-On Testing Exercise: WebGoat – HTTPOnly

8) How to Prevent Clickjacking Attacks

Section Overview: The section introduces another common web application attack most developers aren't familiar with. It explains how and why allowing your content to be framed allows this attack to work and the consequences of such attacks. It discusses the significance of these types of flaws and presents several approaches for how developers can defend their applications against this type of attack.

a) Overview of Clickjacking
b) Clickjacking Defenses
   i) Approach 1: Framebusting Code
   ii) Approach 2: X-Frame-Options Header
   iii) Approach 3: Content Security Policy (CSP)
c) Other Framing Threats
d) Exercises and Labs
   i) Hands-On Demo: WebGoat – Clickjacking

9) How to Architect Input Validation Solutions

Section Overview: The section provides a basis for understanding the important of proper input validation. It walks through common design and implementation approaches to validate user input and discusses the strengths and weaknesses associated with each approach. It starts off with a focus on threats associated with unvalidated user input. It introduces and explains these threats, demonstrates the attacks, and allows students to apply what they have learned by using hands-on lessons. Throughout the section different technology specific protections are explored and discussed as well as the best practices associated with quality attributes of proper input validation.

a) Introduction
   i) Lack of Validation
   ii) Hidden Fields
   iii) Use of Positive Validation
   iv) Regex's
   v) Unchecked Redirects and Forwards
b) How to Validate Outside Applications
c) How to Validate Within Applications
d) How to Respond to Input Validation Issues
e) Validating Data from other Sources
   i) Safe File Uploads/Downloads
f) Exercises and Labs:
   i) Spot the Bug(s): Input Validation Flaws
   ii) Hands-On Testing Exercise: WebGoat – Hidden Fields
   iii) Hands-On Testing Exercise: WebGoat – JavaScript

10) How to Protect Sensitive Data

Section Overview: This section discusses common cryptographic problems associated with web applications as well as caching of sensitive data. It demystifies and dispels the myth that crypto is extremely complex by walking through various simple and straightforward code examples. These code examples are technology specific and include examples of encrypting, decrypting, hashing, and the use of SSL. It also discusses other common flaws that can lead to the exposure of sensitive data.

a) Overview
b) How to Choose the Right Algorithm
c) How to Encrypt, Decrypt, Sign, and Hash
d) How to Avoid Replay Attacks
e) How to Use SSL Sockets
f) How to Protect Sensitive Data in Caches in Applications

11) How to Use Databases Securely

Section Overview: The section provides the material necessary to use a database securely. Threats related to securely connecting to a database, validating input, using SQL, handling errors and logging, and validating results are covered. Some architectural concerns are also discussed in terms of centralizing the security functions related to accessing a database securely.

a) Overview
b) How to Prevent SQL Injection
c) Protecting Database connection Strings (usernames/passwords)
d) Minimizing Privilege

e) How to Handle SQL Exceptions and Verify Results
f) Database Layer Access Control
g) Architectural Patterns for Database Security (DAO)
h) Exercises and Labs:
    i) Hands-On Testing Exercise: WebGoat – SQL Injection

12) How to Handle Errors and Log Security Events

Section Overview: This section introduces the importance of proper error handling and security logging mechanisms for security critical events. Throughout the section technology specific logging APIs and error handling strategies will be introduces and discussed.

a) Overview
    i) Example Real World Fail Opens
b) How to Configure Error Handling
c) Error Handling Best Practices
d) What Security Events to Log and What Data to Capture
e) Detecting and Responding to Attacks
    i) OWASP AppSensor
f) Exercise and Labs:
    i) Hands-On Testing Exercise: WebGoat – Fail Open Authentication Scheme

13) Avoiding the Use of Components with Known Vulnerabilities

Section Overview: This section introduces the importance keeping track of which versions of 3rd party and open source components are being used in a project, and keeping them up to date when new versions become available. Most updates to such components include security updates. Without updating, this introduces significant risk to your applications.

a) Overview
    i) A Huge Risk Most Dev Teams aren't Dealing With
    ii) The Proliferation in the Use of Open Source
    iii) Serious Vulnerabilities in Open Source are Common
b) What Can You Do?
    i) Automation Can Warn You When Your Components Are Out-of-date/Vulnerable
    ii) Develop a Process For Updating Frequently

14) How to Access Services Securely

Section Overview: This section discusses security issues associated with external connections, and walks through various best practices. This section is used as a review of all the practices covered in the course thus far. Students should realize that all the practices they've learned for protection of a web application should apply to an external connection as well.

a) A Pattern for using Services Securely
    i) Vulnerability Examples
b) Applying the Pattern to Prevent Command Injection
c) Architecting Secure Service Access
d) Examples of How to Access Services Securely
e) Exercises and Labs:
    i) Hands-On Testing Exercise: WebGoat – Command Injection

15) HTML5 Security

Section Overview: HTML5 is gaining in popularity and browser support. This section describes a number of security features introduced in HTML5 and how developers can take advantage of them to improve the security of their applications and/or leverage new HTML capabilities in a secure way.

a) What is HTML5
b) HTML5: Local Storage
    i) Security Implications of Using IndexDB
c) HTML5: WebSockets
    i) Using WebSockets Securely
d) Cross-Origin Resource Sharing (CORS)
    i) CORS Preflight Requests
    ii) CORS Security Risks

16) AppSec at DevOps Speed and Portfolio Scale

Section Overview: Software development is moving much faster than application security with new platforms, languages, frameworks, paradigms, and methodologies like Agile and Devops. This section describes how development organizations can instrument their entire IT organization with passive sensors to collect real-time data that can be used to identify vulnerabilities, enhance security architecture, and enable application security activities to generate significant measureable value.

a) Comparison of AppSec to Healthcare
b) Traditional SDLC Approaches
c) Starting Over, at Portfolio Scale, and DevOps Speed
d) Example Sensors
    i) Clickjacking
    ii) Security Headers
    iii) Access Control
    iv) Known Vulnerable Components
    v) CSRF
    vi) Injection
    vii) Architecture, Inventory, More …
e) Building Continuous AppSec Throughout Lifecycle
f) What Sensors Does Your Organization Need?
g) What Security Do You Expect vs. What Are You Actually Measuring?
h) Aligning Sensors with Business Concerns
i) Developing a Portfolio Wide Dashboard

17) References

a) Books
b) OWASP Resources
c) Microsoft Application Security Resources
d) Web Application Security Consortium Guidelines

18) The Challenge

Section Overview: The challenge section allows students to step back, look at what they have learned and apply this knowledge by performing a final hack on the hands-on Challenge lesson. This lesson combines many of the vulnerabilities previously discussed into a single lesson (with multiple stages). This lesson doesn't contain any hints, as do previous lessons. In previous lessons, hints are included to guide students through each stage of an attack. While the instructor assists students, this is the time to allow the students to use their creativity and the knowledge they have gained from this course to successfully compromise the final lesson.

a) Exercises and Labs:
   i) Hands-On Testing Exercise: WebGoat – Challenge Stage 1 – Break Authentication
   ii) Hands-On Testing Exercise: WebGoat – Challenge Stage 2 – Steal the Credit Cards
   iii) Hands-On Testing Exercise: WebGoat – Challenge Stage 3 – Deface the Web Site

1 Professional Services Introduction

Attacks at the application level have become the number one threat to web based applications according to Gartner and Mitre. The impact of these attacks can range from exposure of Personally Identifiable Information (PII) to interruption of application availability to fraudulent use of an application. Guarding against these attacks requires applications to be built with security mechanisms designed to stop application attacks.

Aspect Security (www.aspectsecurity.com) has developed its Application Security Verification Services to verify that the appropriate security mechanisms are in place and functioning properly. Application Security Verification Service combines security code reviews and penetration testing to provide the assurance required for critical applications.

The goal of the security verification is to provide coverage for the most common web application vulnerabilities including all 10 vulnerability areas described in the Open Web Application Security Project's (OWASP) (www.owasp.org) "Ten Most Critical Web Application Security Vulnerabilities". The Standard verification addresses the following topics: Authentication, Access Control (Authorization), Parameter Use, Cross-Site Scripting, Cross-Site Request Forgery, Command Injection, Error Handling, Cryptography, Denial of Service, System Administration, and Server Configuration.

2 Aspect Security Credentials

Aspect's senior management team has been successfully providing both commercial and government clients with solutions to complex security problems for over 20 years. Through a wide range of engagements and positions held with well-respected organizations within the security community, Aspect's engineers have provided clients with support in the areas of web application security, network security, secure software development, security architecture and design, security policy development, security integration, and system assessment and certification services.

2.1 Corporate Background

Immediately prior to founding Aspect, the organization's principals were largely responsible for establishing the security standards and services practice for one of the world's largest commercial IT security organizations as well as forming and running their web application security services group. In this capacity, Aspect's engineers provided security solutions addressing the complex web hosting requirements of many of the Fortune 500. Since its founding in Q1 2002, Aspect has been focused on assuring the security of a wide variety of traditional, non-traditional, and complex applications within verticals such as the financial, healthcare, insurance, and pharmaceutical communities, as well as the US government. Our staff of senior application security engineers have focused on the problems associated with developing and fielding secure products and applications for over a decade.

A well respected player in the application security niche, Aspect leads the Open Web Application Security Project (OWASP) "Top Ten" project which published the Top Ten Most Critical Web Application Security Vulnerabilities. The OWASP Top Ten list was created to focus government and industry on the most serious web application security vulnerabilities. This resource has been seen as a critical development for consumers and vendors alike, and has been adopted by the Federal Trade Commission, the Defense Information Systems Agency, and incorporated into the Payment Card Industry Data Security Standard (PCI DSS) as a web application security standard.

One of Aspect's key differentiators is our level of expertise in both security and software development. In addition to having a comprehensive understanding of the security risks facing organizations today, our engineers also possess broad experience with variety of development methodologies including both government and commercial procurement and development efforts. Aspect encourages the engineering team to continually expand their base of knowledge, and as a result most staff members possess one or more advanced degrees as well as multiple security certifications (i.e., CISSP, SANS GSEC, CISM). Aspect's principals are seen as thought leaders within the application security industry and have published several dozen papers within refereed security journals and conferences. Committed to continuous process improvement, Aspect's founders authored the System Security Engineering Capability Maturity Model (SSE-CMM). Seen as the

foundation reference on security engineering process improvement, the model was recently accepted as ISO standard 21827.

Aspect is a highly trustworthy organization and has performed classified projects for various government agencies for over a decade. Aspect's engineers have been through appropriate background investigations. We carefully protect all customer sensitive information while it is in our custody.

2.2 Relevant Experience

Our team is uniquely qualified to support Client in their efforts to identify vulnerabilities associated with their custom application. Our team will be staffed with experienced security engineers who know the types of mistakes that are frequently made in the development and deployment of applications, products and systems. We have extensive experience identifying vulnerabilities within a wide range of software implementations including COTS products and custom applications using a variety of analysis techniques.

Prior engagements demonstrating our expertise and capabilities include the following (names have been withheld at request of clients):

Army: Aspect developed a code review standard and process in support of the Army's deployment of an Enterprise Resource Planning System. In addition to training the development team, Aspect led the code review of 5 million lines of Java and ABAP (SAP) code utilizing a mix of client and Aspect resources.

Financial Organization 1: Within this engagement Aspect successfully reverse engineered an automatically downloaded client side custom applet supporting a large currency trading application. Aspect was able to circumvent the cryptographic controls in place to protect data by modifying the applet and forcing it to write data to a file prior to the encryption module being invoked.

Financial Organization 2: In this situation, the target was a bond trading application. Aspect engineers once again reversed engineered a custom client downloaded by the application. Even though client to server communications were protected through the use of a custom protocol Aspect was able to modify the output of the client which led to the compromise of the server and back end database.

Large Global Bank: Aspect has provided ongoing security testing support to this organization's internal development team. The J2EE applications tested support the business' main web-facing applications. In addition, Aspect has conducted black box web application and network penetration testing to identify weaknesses that could leave critical applications and their supporting infrastructure susceptible to attack from either external or internal entities.

International E-Commerce Sites: Aspect conducts annual application security assessments on this client's US and European e-commerce applications. Significant issues in almost all areas of the OWASP Top Ten have been identified in this customer's sites and presented to the customer along with specific recommendations for remediation.

International Commercial Real Estate Management Firm: Aspect performed a security review of an application supporting critical business functions such as lease management. The review included code review, scanning, and penetration testing techniques and helped to alert the client to several areas of potential vulnerability requiring immediate attention.

General Electric: Aspect has provided code review and application penetration testing services to many different business units within General Electric. These assurance activities represent key steps in GE's process for determining if an application has been properly secured before it is allowed to go operational.

Fortune 50 Organization: Aspect engineers have reviewed and performed assessments on the web interfaces of a number of commercial products as part of the organization's due diligence process. These products have included an e-learning product, an electronic billing product, an investment survey portal, a product from Lotus that tracked business opportunities, and a large CRM product.

Nintendo of America: Aspect has provided a wide range of application security services to Nintendo in support of a high traffic J2EE e-commerce application. The services provided include developer training, security policy development, code review and application penetration testing.

3 Application Security Verification Methodology

Aspect has developed a unique application security verification methodology that combines vulnerability scanning, code review, and penetration testing in a single solution. Essentially, we use the strengths of each approach to thoroughly examine the entire application in a more efficient manner than any one approach on its own. The net result is a very effective security review in a short amount of time. Aspect proposes applying this methodology to conduct a security verification of Client's application.

While we believe the combined approach described here is the most cost-effective approach for finding vulnerabilities in an application, Aspect will provide penetration testing or code review services separately as well. Note that our level of effort is the same whether we do penetration testing only, code review only, or any combination of those activities.

3.1 Kickoff

The engagement will initiate with a brief 1-2 hours kickoff meeting conducted via conference call where key Customer personnel will provide Aspect engineers with an overview of the application architecture and functionality. Prior to this meeting Customer will provide Aspect with any relevant and available background documentation on the application as well as the source code. Aspect can perform the security verification more quickly and accurately with information about security policy, requirements, design, and implementation, if this information can be made available. Any build or install documentation (which describes the organization of the code) is also helpful. A secure method of transmission will be established to support the transfer of documentation and source code.

3.2 Web Application Security Verification

We begin the verification by performing an application vulnerability scan (where applicable) and conducting a walkthrough of the application. The purpose of the walkthrough is to understand the application's intended operation and to identify security flaws. During this step we begin to track the different components of the application and formulate possible areas of weakness. We do not attempt penetrations at this point, although we keep a careful list of possible targets. We then proceed to examine the code by following the threads for the major security functions. As we uncover items, we add them to the list for prioritization. Each issue is carefully documented with code fragments and details on exploitation. These issues also feed into a list of potential penetration test attempts.

Over years of code analysis, we have developed a proprietary methodology that improves on all other methods of finding security flaws in source code. We have developed an extensive database of common library calls and methods that are difficult for programmers to use securely. The database also includes methods that are commonly used in security mechanisms such as encryption, authentication, logging, and error handling.

We use a suite of tools and techniques that allow us to find the use of these calls efficiently throughout a body of code. Using this approach, we can quickly find candidate security vulnerabilities, and our experts can analyze them to validate whether the code is actually vulnerable, or if the developer has taken proper protections. For example, using our approach, we can ensure that an application is properly protected against SQL Injection. First, we locate the use of database-related methods. Then we trace all the users of those methods to ensure that they have taken proper precautions by validating and sanitizing the input parameters. If they haven't, we continue to check callers or decide that we have found a vulnerability.

While we analyze source code, we also prepare a list of areas most likely to result in a successful penetration. This list is prioritized based on the likelihood of a successful compromise and the magnitude of the expected consequence. During penetration attempts we focus on finding the highest risk items first, to perform the most efficient penetration. As we proceed through our prioritized list of potential weaknesses, we generate a detailed report of findings. If we uncover another likely penetration candidate during the exploitation process, we add it to the list and reprioritize. Our primary goals for the review are completeness and rigor. We are careful to ensure that all areas of an application are tested, and we do not stop after finding a certain number of flaws.

3.3 Documentation of Results

A final report will be prepared documenting the results of the security verification. The report will describe the work performed, what was discovered, and will provide specific recommendations for improving the security of the application and any recommendations for further activities. The report will highlight:

Security practices inconsistent with Client's security goals,

Software, system and application vulnerabilities discovered,

Details on the severity, likelihood, and consequences of each vulnerability,

Examples of the identified vulnerabilities, and

Specific recommendations for addressing each vulnerability.

During the analysis and testing, each finding is documented immediately when it is uncovered. The finding is reviewed by at least one other member of the team upon its completion. When the analysis and testing is complete, the report is generated from the database of findings. This report is reviewed in its entirety to ensure that all the findings are clear, detailed, and provide the best remediation recommendations possible.

This report includes both the technical details of the results as well as an executive summary and ranked list of the most significant issues identified. A sample security verification report has been included for Client's review.

3.4 Remedial Review

At customer's option, Aspect will perform a remedial review on the fixes applied to the vulnerabilities identified in the original verification. The deliverable from this review will be an update to the original report detailing the effectiveness of the fixes.

4 Application Security Areas of Focus

Using the methodology described above, Aspect will focus on those areas where we most often find security problems in applications. These areas typically include common security mechanisms like authentication, access control, logging, and encryption. We also search for common vulnerability areas such as buffer overflows, cross-site scripting, SQL injection, parameter validation, thread safety, and denial of service. A description of the areas covered can be found in the table below.

Areas	Description
Authentication	Properly validating the identity of site users is complex and is a prime target for attackers. We analyze your authentication scheme to determine whether spoofing, corruption, or brute force attacks might gain unauthorized access.
Access Control (Authorization)	We check the mechanisms that enforce the access control rules that define which users can access which parts of the web application. We verify that users cannot access parts of the site that they should not be allowed to access.
Parameter Use	Attackers frequently modify HTTP request headers, form fields, URL parameters, and cookies in an attempt to force an application to misbehave. We verify that proper input checking is performed to prevent malicious parameters from causing your web application to misbehave.

Areas	Description
Cross-Site Scripting	Web applications can be used as a mechanism to transport scripting attacks to another user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user. We search for unchecked input which could pass embedded attack scripts through your defenses.
Cross-Site Request Forgery	A Cross-Site Request Forgery (CSRF) vulnerability is a flaw in a web application that accepts an unauthorized transaction from a victim tricked into submitting it. This attack relies on the fact that a victim's credentials are usually automatically submitted by the victim's browser with each request to a site.
Injection	Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the end system may execute those commands on behalf of the web application. There are three main endpoints for such attacks: **SQL Injection** - Backside databases used by web applications frequently hold the most sensitive data. We verify whether parameters are validated before being used in queries. **Operating System Injection** - Many web applications invoke libraries, scripts, and operating system utilities to perform certain functions. We determine whether these libraries could be tricked into doing things that the developers did not intend. **Other Interpreters** - Many web applications access external resources such as LDAP, mail, FTP, web services, and XML data sources. Attackers will attempt to force your web application to misuse these services. We check these calls to make sure they properly validate all input they use to ensure they cannot be exploited.
Error Handling	Flaws in error handling mechanisms can easily cause web applications to crash and can also reveal sensitive information about how the application functions. We verify that error messages do not reveal sensitive internal information that can help hackers find security flaws in the site's security mechanisms.
Cryptography	Web applications frequently use cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection. We search for improper or inconsistent use of cryptographic mechanisms.
System Administration	All web applications need to be administered, and many have their own administration functions built in. We search for these functions and verify that no one can misuse or usurp their power.
Server Configuration	Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box. If we are provided configuration data, or access to the actual servers, we search for configuration flaws and the presence of default accounts and unpatched vulnerabilities.
Denial of Service	Web applications are easy denial of service attack targets, but there are techniques to defend against them. We examine your application to determine its susceptibility to this form of attack.

Application Security Areas of Focus

The web application review focuses on discovering vulnerabilities in Client's custom application software. We do not attempt to identify vulnerabilities in the host operating system, other host software, or network infrastructure. The results of our service depend on information from the customer, depth of analysis requested, and access to the application. We do not guarantee that all security issues with the application will be identified or that an application is secure or error free.

5 Technical Tools and Utilities

Aspect uses a wide variety of tools to find vulnerabilities in web applications. However, we recognize that automated testing only addresses a relatively small percentage of the possible security issues in a web application. Many simple security problems, including the well-publicized Microsoft Hotmail/Passport issue, would never have been uncovered with an automated approach alone. Therefore, we use tools that leverage our long experience testing web applications and reduce the time required by our experts to uncover and validate a vulnerability.

Source Code Analysis

AspectCheck – Aspect's proprietary tool for finding and tracing the use of potentially dangerous methods throughout a body of code. If source code can be made available, this analysis produces far more comprehensive results than other methods of finding vulnerabilities.

Eclipse – We have customized the Eclipse IDE with a set of plugins and configurations that facilitate code review. We use this environment to browse source code and analyze security.

Commercial Code Analysis Tool – On occasion, we will employ a commercial static analysis tool to support the process of locating and categorizing vulnerabilities in application source code. Such tools are used based on the nature of the application and the languages employed.

Penetration Testing Tools

Burp Pro – A web application proxy is used to facilitate the testing effort, giving the analyst the ability to intercept and modify web application and web service traffic. Burp Pro includes many specialized tools for testing various aspects of application security.

SOAP UI – A web services testing framework

Port and Vulnerability Scanning Tools

Qualys – A commercial scanning service that can scan internet accessible applications for vulnerabilities.

Nmap – A network mapping and port scanning tool.

Nessus – A vulnerability scanner with a wide variety of NASL scripts designed to detect flaws.

Commercial Vulnerability Scanner – On occasion, we will employ a commercial vulnerability scanner to find configuration issues in the platform. Such tools are used based on the details of the application deployment.

6 Security Verification Report

A final report will be prepared documenting the results of the security verification. The report will describe the work performed, what was discovered, and will provide specific recommendations for mitigating areas of risk that are identified. The report will highlight:

Security practices inconsistent with the Customer's security goals,

Software, system and application vulnerabilities discovered,

Details on the severity, likelihood, and consequences of each vulnerability,

Examples of the identified vulnerabilities, and

Specific recommendations for addressing or mitigating each vulnerability.

This report includes both the technical details of the results as well as an executive summary and ranked list of the most significant issues identified.

7 Contract Information

Aspect Application Security Verification services are available via Schedule Contract Number GS-35F-0082U, awarded under solicitation number FCIS-JB-980001-B for SIN 132-51, extended in 2012, and modified in March, 2015. For more information please send email to info@aspectsecurity.com or call 301-604-4882.

Contrast

Applications Test Themselves Continuously, without Experts!

Contrast Makes Applications Test Themselves

Even with existing tools and experts, securing applications against hackers is a manual, time consuming and expensive process that simply can't meet the scale and speed of modern software delivery practices. Contrast is a software product that quickly and accurately pinpoints security flaws in running applications, eliminating the long delays and high costs associated with legacy SAST and DAST solutions. Contrast installs and starts working continuously in minutes, requiring no experts, scripts, tuning or testing. Whether securing one application or thousands, Contrast makes applications test themselves anywhere they run, so no one has to.

How does it work?

Contrast is an enterprise-class application security solution. Distributed Contrast software agents continuously monitor code execution and data flow of running applications for dangerous behaviors that make them vulnerable. This approach not only finds issues faster and more accurately but it delivers to developers the complete code level evidence and expert guidance that accelerates remediation. Findings are sent to bug tracking systems and/or Contrast's centralized console, from which application security personnel track live views, receive alerts and generate reports for their entire application portfolio.

Why Contrast?

Contrast Security has been helping enterprises and governments produce secure code and dramatically reduce application security risks for over 10 years. That unique expertise led to Contrast's patented technology, recognized by experts, analysts and some of the world's most security-conscious organizations as not just visionary, but highly effective.

Supported Technologies

Contrast works on the most widely deployed enterprise application technologies on the market today including Microsoft .NET and Java. Check www.contrastsecurity.com for the latest list of supported platforms.

Contrast Features

Real-Time Vulnerability Detection and Expert Guidance

Contrast monitors Java and .NET code execution, data flow, configurations and more to quickly find dangerous vulnerabilities with virtually no false positives. Line-level pinpointing eliminates guesswork while context sensitive guidance enables quick remediation.

Library Inventory and Analysis

As much as 80% of software code comes from open source and third-party libraries. Contrast discovers third-party libraries and the known (and unknown) risks they bring.

Enterprise Portfolio-Class Scalability

Contrast transparently automates application security to support portfolios of virtually any size. New applications are discovered automatically as they are run. Executive-level portfolio dashboards display the entire portfolio security posture in real-time.

Agile Speed and Seamless Automation

Continuous integration and deployment require continuous security. Scriptable silent installers, automated updates, and a REST API enable Contrast to deliver security as fast as applications change.

SaaS, On-Site and IDE Deployment

It takes minutes to go from zero to resolving application security issues using Contrast's SaaS service and Contrast for Eclipse for Java developers. Contrast can easily be hosted and administered on-site as a private service enabling a completely and easily administered private service.

Latest Information

For the latest information on Contrast and the technologies it supports, please visit: www.contrastsecurity.com.

Application Security eLearning Catalog

Aspect Security has a growing catalog of over 50 eLearning modules that teach developers, software testers, architects, application security specialists, and IT executives the most common security risks in custom developed software, and most importantly, how to avoid them in a cost effective manner. These modules, which average about 20 minutes in length, cover the entire gamut of the application security vulnerability landscape, including the OWASP Top 10 and SANS Top 25 lists of most common software vulnerabilities. The following provides more details on our course catalog and provides a specific list of all the modules currently in our catalog today as of May 2015. Please check online for the most current information: http://www.aspectsecurity.com/elearning.

The above summary of Aspect's eLearning curriculum is also available in brochure form that you can download from: http://cdn2.hubspot.net/hub/315719/file-1890307152-pdf/download-files/eLearning_Data_Sheet_2014_Web_Version.pdf

Aspect currently has the following catalog of eLearning modules. Please check: http://cdn2.hubspot.net/hub/315719/file-1890307167-pdf/download-files/eLearning_Curriculum_2014_Web_Version.pdf to get the most up to date catalog of our eLearning courseware.

Application Security Awareness Series

Provides training about application security and covers fundamental application security principles. Each module covers specific security controls such as: Input Validation, Access Control, Authenticating Users, Protecting Sensitive Data (and more) giving learners a solid foundation in application security principles.

100 Introduction to Application Security *[12:01 minutes]*

Discusses the risks associated with software applications and suggests the outline of a practical application security program. We focus on enabling you to cost-effectively produce and deploy secure applications. Additionally, this module covers the common challenges an organization faces when it starts to address application security.

110 Input Validation *[15:57 minutes]*

We define input validation, provide basics for verifying whether an application is vulnerable to input validation attacks, and discuss common techniques for defending against these. In order to get the most out of this module, students should be familiar with the basic workings of web applications, including HTML and HTTP.

120 Access Control *[18:41 minutes]*

We discuss how to limit the access of authenticated users to the resources and functions in your web application. Basic techniques are provided to verify that an application is free from access control vulnerabilities. Techniques for defending against these attacks are also provided. To get the most from this module, students should understand the basics of authentication.

130 Authenticating Users *[27:26 minutes]*

User authentication is defined, common attacks and vulnerabilities are discussed, and strategies to defend against attacks and avoid vulnerabilities are provided. Credentials, cookies, sessions, and user management are covered. To get the most out of this module, students should be familiar with the basics workings of web applications, including HTML and HTTP.

140 Error Handling and Security Logging *[14:08 minutes]*

Learn about basic error handling patterns to prevent information leakage and denial of service. Proper logging techniques are described to ensure a complete security record. We show you how to implement simple intrusion detection techniques to make your applications more resistant to attack.

150 Protecting Sensitive Data *[17:11 minutes]*

Basic techniques for protecting particularly sensitive or regulated data, such as credit card information, social security numbers, and healthcare data are addressed. We show you how to verify that your applications properly protect any sensitive data that they process. The basics of key management, encryption algorithms, hashing, and secure random number generation are covered.

160 Securing Communications *[19:22 minutes]*

Security concerns related to transporting data across a network, both on the front end and with backend services are discussed. We introduce basic security controls for securing these connections, including: SSL, authentication, access control, and data encryption. To get the most out of this module, students should be familiar with the basic workings of web applications, including HTML and HTTP; and have an understanding of the basics of session management.

180 Buffer Overflows *[To be released Q1 2015]*

We define Buffer Overflows that are sometimes called "Buffer Overruns." Guidance is provided to verify whether an application is vulnerable to buffer overflow attacks, and common techniques for defending against these types of attacks. In order to get the most out of this module, students should have a basic understanding of software development practices.

Secure Software Development Series

Covers the OWASP Top Ten 2013 and fulfills the training requirement mandated by PCI/DSS. Vulnerability areas such as Cross-Site Scripting (XSS), SQL Injection, Clickjacking, and more are covered. Specific vulnerability prevention and removal techniques for each security/ vulnerability area are included. We teach the most cost-effective testing, prevention and remediation techniques. Specific platforms and technologies are addressed such as Java, .NET, Mobile, AJAX, Rich Internet Applications and Web Services.

Validating User Input

211 Input Validation Strategy *[30:42 minutes]*

We provide an overview of basic defenses against input manipulation and injection attacks. Strategies discussed include: minimizing input to the application, validating incoming data, and sanitizing outgoing data. Pitfalls of input validation such as relying on client-side validation, missing canonicalization, and inconsistent application of input validation, provide students with the right information to begin exploring new strategies. To get the most out of the module, students should be familiar with the basic workings of web applications, including HTML and HTTP.

212 Preventing Injection Attacks *[20:28 minutes]*

Students are introduced to injection flaws, examples of common injection attacks, and basic defenses against injection. Key concepts such as: interpreters, common injection vulnerability patterns, injection attacks, recommended defenses against injection attacks, utilizing static commands, and more are explained. Basic knowledge of SQL, HTML, XML, and command line shells is helpful but not required.

213 Preventing XSS Vulnerabilities *[20:56 minutes]*

This module details exactly how all the different types of Cross-Site Scripting (XSS) vulnerabilities work and how they can be exploited. We demonstrate how to find XSS flaws and evaluate their exploitability. Students learn about prevention strategies for XSS by seeing how to properly escape output in each of the different contexts in HTML.

214 Using Canonicalization and Encoding *[23:22 minutes]*

This introduction to canonicalization and encoding covers why proper decoding and encoding is critical to performing effective input validation and identifying attacks. To get the most out of this module, students should be familiar with the basics of input validation and defending against injection attacks.

215 Performing Secure File Uploads and Downloads *[12:11 minutes]*

All of the complexities involved with implementing file upload and download securely in a web application are addressed. Validating upload requests, storing files carefully, and testing file and upload features for vulnerabilities are covered. Students should be familiar with HTTP and basic input validation techniques.

216 Preventing Header Injection *[18:06 minutes]*

Problems related to HTTP header injection and their consequences are presented. We discuss techniques for verifying whether an application is vulnerable to header injection and provide techniques for defending against these attacks. To get the most out of this module, students should be familiar with HTTP and injection techniques.

226 Unsafe Redirects and Forwards *[17:29 minutes]*

Common security weaknesses related to sending redirect responses through the browser, such as using untrusted information in the location, and disclosing sensitive information are illustrated. Access control concerns associated with performing server side forwards are covered. Students should understand the basics of HTTP and input validation.

331 Understanding DOM-Based XSS *[14:38 minutes]*

Cross-Site Scripting (XSS) is a prevalent vulnerability. Here we take a look at a specific type, DOM-Based XSS. We describe how to recognize this vulnerability and prevent it from appearing in your code. To maximize your understanding of this module, you should be familiar with Reflected and Stored XSS.

Controlling Access

221 Access Control Strategy *[16:56 minutes]*

Approaches for defining and enforcing access control policies are introduced. Core concepts of access control are discussed and applied to typical web application architectures. In particular, enforcing access control at the URL, business logic, data, and presentation layers are evaluated. Techniques for verifying application access control implementations are also discussed. To get the most out of this module you should be familiar with the basics of authentication.

222 Presentation Layer Access Control *[12:35 minutes]*

Understanding that presentation layer access control does not provide protection against attack is a critical concept for developers. In this module, techniques for validating access control demonstrate that proper controls must be in place in addition to the presentation layer. Topics covered include forced browsing and direct object references. To get the most out of this module, you should be familiar with access control strategy.

223 URL Access Control *[19:10 minutes]*

Effective strategies for performing access control at the URL layer are presented. Techniques are demonstrated for verifying your application's URL based access control implementation and ensuring that it is robust against well-known common weaknesses. Common attacks like forced browsing are shown. To get the most out of this module, you should be familiar with HTTP and access control strategy.

224 Business Layer Access Control *[17:26 minutes]*

Techniques to enforce access control for business functions are presented. Students will learn how to verify business layer access control through testing and code review. In addition the module discusses implementing business layer protections in a simple, structured way. To get the most out of this module, you should be familiar with access control strategy.

225 Implementing Data Layer Access Control *[12:02 minutes]*

We discuss how to secure the sensitive data that is stored by your application so that it is safeguarded from attackers. Data access control is a security mechanism that ensures that users are only allowed to access authorized data based on the user's identity, roles, and/or permissions. Common attacks on application data and strategies for defending against them are discussed. Students should be familiar with access control strategy.

227 Clickjacking *[14:03 minutes]*

The common attack known as Clickjacking, is when attackers frame pages from other sites and trick users into unwittingly clicking on those pages. The module discusses the various approaches for defending against Clickjacking by preventing your pages from being framed, including framebreaking scripts and the use of new headers, such as X-FRAME-OPTIONS.

Authenticating Users

231 Authentication Strategy *[17:26 minutes]*

We illustrate the various techniques used by web applications to authenticate users and handle identity. The strengths and weaknesses associated with different authentication schemes are examined and recommendations are provided to minimize authentication vulnerabilities. Also discussed are techniques for verifying authentication schemes. To get the most out of this module, students should be familiar with the basic workings of web applications, HTTP, and sessions.

232 Understanding HTTP Authentication Schemes *[17:50 minutes]*

This module delves into detail about BASIC authentication, form-based authentication, and the proper use of sessions. Prevalent attacks on these authentication schemes, such as brute force attacks, session hijacking, and session fixation are discussed. Students will learn several best practices to protect authentication schemes from attack. Students should understand HTTP and sessions.

233 Secure Session Management *[16:36 minutes]*

Session hijacking is a critical security vulnerability as it allows user sessions to be completely taken over by an attacker. Students will learn techniques to determine whether applications are vulnerable to session attacks and how to defend their code against these issues. To get the most out of this module, students should be familiar with HTTP and authentication strategy.

234 Protecting Credentials *[17:43 minutes]*

Students get an in-depth understanding about the issues surrounding protecting authentication credentials that are used by users, applications, and even systems. We show you techniques to attack, verify and defend your application against improper credential protection.

235 Managing Identity within an Application *[13:03 minutes]*

Applications need to have access to information about users, update that information, and occasionally remove those users. In addition, the current user identity is used in many security controls including access controls, logging, and more. This module discusses the management of identity within an application and provides details to implement identity properly. Students should have a basic understanding of authentication strategy to get the most out of this module.

236 Preventing Forged Requests (CSRF) *[14:17 minutes]*

Cross-Site Request Forgery (CSRF) allows an attacker to trick a victim's browser into issuing authenticated requests to a vulnerable web application. This module introduces CSRF and discusses basic techniques for checking applications for CSRF vulnerabilities. Basic defenses against this common vulnerability are provided. To get the most out of this module, students should be familiar with the basic workings of web applications, including HTML and HTTP and must also understand the basics of authentication and session management.

Error Handling and Security Logging

241 Handling Security Errors and Detecting Attacks *[11:20 minutes]*

Specific guidance on how to handle security exceptions, and others exceptions securely and how not to leak implementation details to attackers is provided. We also cover establishing a security exception hierarchy and a general error handling scheme. Strategies for generating safe error messages and detecting intrusion are provided.

242 Effective Security Logging *[15:30 minutes]*

Details how to create effective security logs that can be used to identify and triage attacks are discussed. We cover which events to log, what to log with each event, and strategies for ensuring that proper logging occurs. Ensuring accountability, log usage for forensic purposes and techniques for verifying logging implementations are discussed.

Protecting Sensitive Data

251 Introduction to Cryptography *[15:48 minutes]*

Cryptography basics are presented including: the fundamentals of using keys and algorithms to securely encrypt data, key management, hashing, digital signatures, and random numbers. This module provides a foundation for the "Using Cryptography Securely" module.

252 Using Cryptography Securely *[16:04 minutes]*

The basics of using cryptographic techniques to encrypt and hash data for security purposes are addressed. We focus on using standard approved cryptographic libraries securely, and discourage the creation of custom cryptographic code. Common vulnerabilities associated with the use of encryption are discussed, including credential handling, failure to encrypt sensitive data, algorithm choice, and more.

253 X.509 Certificates *[25:35 minutes]*

X.509 Certificates are the most common type of digital certificate. You will learn how to describe the structure and key elements of X.509 certificates, understand the Key and Certificate lifecycle, and install and properly protect certificates and more.

261 Using SSL *[18:07 minutes]*

Students will learn how to use SSL properly to protect data in transit. We explain how SSL works and the many ways that it can be used incorrectly that open communications to attack. The transitions between secure and insecure parts of a website are carefully examined. Techniques for verifying that SSL has been configured correctly in an application are presented. To get the most out of this module, students should be familiar with the basic workings of HTTP, cryptography, and certificates.

262 Securing Cookie Use *[21:38 minutes]*

A cookie is a short text string sent by web applications to maintain state in the browser. Protecting this state and understanding the limitations of cookie protection are critical to web application security. This module covers common attacks against cookies and details the use of protection mechanisms such as the 'Secure' and 'HttpOnly' flags as well as encryption and integrity seals. To get the most out of this module, you should be familiar with the basic workings of web applications, including HTML and HTTP.

263 Using Services Securely *[20:57 minutes]*

We discuss the risks and necessary controls to use backend services from a web application securely. Topics include the importance of securing services, handling authentication and access control with services, and performing input validation with services. We explain how to securely transmit and store data via services and verify that service use is secure. Before taking this module, students should have some background in security fundamentals.

264 Using SSL with Java & .NET *[22:48 minutes]*

The relationship between server and client side certificates, and private keys are explained. Students will be introduced to commonly used, publicly available SSL libraries for Java and .NET and how to properly use them.They will also learn how to install and protect client and server side certificates, and private keys in Java and .NET.

Platform Specific

281 Introduction to Secure Coding for Rich Internet Applications *[13:31 minutes]*

Security threats to all the stakeholders in an RIA and the principles for deploying safer code are explained. The need to get security right in the next generation of applications, how to get a hard start on getting security right before all web apps are transformed into RIAs is presented. Students should have a basic background in RIA technology and foundational knowledge of security.

282 Introduction to Secure Coding for AJAX Applications *[15:54 minutes]*

There are complexities securing applications that use AJAX. Web 2.0 client side code frequently exposes vulnerabilities when developers don't remember that the client is under the attacker's control. To get the most out of the module students should be familiar with the basic workings of AJAX applications, including XHR and HTTP.

283 Introduction to Secure Coding for Java EE Applications *[17:36 minutes]*

Secure coding for Java developers is introduced by focusing on three key areas: the need for security controls, the five most critical security areas for Java applications, and how to verify the security of those applications. To get the most out of this module, students should be familiar with the basic workings of web applications, including HTML and HTTP. Knowledge of Java web technology is required.

284 Introduction to Secure Coding for .NET Applications *[26:35 minutes]*

Secure coding for .NET developers, the module discusses the risks associated with .NET software applications, basic methods of secure coding, and testing applications. The three key areas of focus are: the need for security controls, the five most critical security areas for .NET applications, and how to verify the security of those applications. To get the most out of the module students should be familiar with the basic workings of web applications, including HTML and HTTP. Knowledge of .NET web technology is required.

Web Services

291 Introduction to Web Services Security *[17:41 minutes]*

Web services, common web service architectural styles, and the security standards available to each style of web service are introduced. WSDLs and the security issues surrounding their use and dissemination are explained in detail. Recommended architectures for providing security services within a web service are provided. To get the most out of this module, students should be familiar with the basic workings of web applications, and web services including XML and HTTP. An understanding of the basics of authentication, access control, and session management are required.

292 Web Services Authentication and Authorization *[20:03 minutes]*

Focusing on the challenges of authenticating and authorizing web service requests, authentication models are discussed, including the use of WS-Security, SAML, and non-standard approaches. Various approaches for implementing access control over web services interfaces from the WSDL through the business logic are presented.

Secure Architectures and Threat Modeling Series

Security architects, development architects, and other specialists can take the following modules to augment their skills and learn how to create secure application architectures.

170 Hardening Application Platforms and Frameworks *[27:33 minutes]*

Techniques to harden application platforms and frameworks and verify that they are secure are presented in this module. The module addresses the application framework, application server, web server and host layers. In order to get the most out of this module, students should be familiar with the basic workings of web applications, particularly in the deployment environment.

271 Hardening Web Servers *[11:51 minutes]*

The necessary steps are explained how to lock down web servers. Versions and patching, default file handlers, connectors, default services, error handling, SSL and certificates, and extension handling are addressed. Masking details that are used to fingerprint servers are explained.

272 Hardening Application Servers *[09:07 minutes]*

Actions required to harden a web application server, including configuration options such as timeouts, ports, services, error handling, connectors, default applications, extension handling, and MIME types are addressed and explained.

273 Using Components Securely *[11:29 minutes]*

The use of components during application development has surged in recent years. We look at the security ramifications of this increased use of libraries and a few things that you can do in your software development process to help minimize these risks. Topics such as searching applications for components with known vulnerabilities, developing an inventory of component use across applications, and choosing a strategy for updating components with known vulnerabilities are discussed.

311 Understanding Mobile Application Threats *[13:00 minutes]*

This module covers the new risks associated with mobile platforms. With close to 5 billion mobile devices in operation, the very concept of an "app" is changing. You'll learn the threats to mobile computing and how to apply existing application security principles in this new environment. We explore new vectors like location-based attacks, SMS, Bluetooth, contacts, photos, purchasing, and calls. Designed for mobile developers, testers, architects, this module will get you started in managing the security of your mobile applications.

420 Threat Modeling and Security Architecture Review *[17:29 minutes]*

Threat modeling and security architecture reviews are efficient and effective methods to identify vulnerabilities throughout the software development lifecycle. We provide a framework to organize components, connections, security controls, threat agents, sensitive assets, and important business functions in order to make security visible. To get the most out of this module, students should be familiar with the general application security areas and associated attack vectors. Students should also have a high-level understanding of security verification processes and techniques.

Application Security Testing Series

Security testers and other specialists can take the following modules to further develop their application security testing skills.

101 Introduction to Application Security Verification *[10:06 minutes]*

Risks associated with software applications and basic testing methods are introduced. Application security verification is the process of ensuring that an application or group of applications properly use the appropriate security controls and do not contain vulnerabilities. This module focuses on introducing what to look for and how to find it.

410 Security Verification Methodology *[13:22 minutes]*

We describe various techniques for verifying the security of an application. We compare automated tools versus manual approaches. Static, Dynamic and Runtime tools are discussed. The module explains how to mix and match these techniques to verify the security of an application efficiently and cost-effectively. Students will learn what's involved in scoping reviews, rating risks, and writing findings. Students should have a working understanding of common application security issues.

411 Effective Security Testing *[13:49 minutes]*

A framework for scoping, structuring, executing, and documenting an application security test is outlined. Effective security testing means getting broad coverage over the application's security controls in as short a timeframe as possible. A proven, repeatable process for getting the most assurance possible out of a security testing program is provided. We discuss where tools can be leveraged. Students should understand common vulnerabilities and basic risk management to derive the most from this module.

412 Effective Security Code Review *[19:42 minutes]*

The goals of an effective code review are defined and guidance is given about how to plan a tailored code review. You will learn how tools complement the manual process. Strategies to begin an effective review of large-scale enterprise applications are presented. To get the most out of this module, students should also have completed prior modules on General Authentication, Access Control, and Input Validation Strategies or have an advanced understanding of this subject matter.

Leaders and Managers Series

Leaders and Managers of application development and security testing teams can take the following modules to augment their knowledge as they serve as application security leaders in their organization.

102 Introduction to the Secure Development Lifecycle *[12:48 minutes]*

Approaches that organizations can use to integrate security into their software development lifecycle (SDLC) are discussed. Key foundations and activities that development teams can use during design and development to produce secure code are introduced. Topics include security architecture, threat modeling, standard controls, standards, guidelines, and secure delivery.

103 Introduction to Managing Application Security *[23:02 minutes]*

Building on over a decade of experience working with major corporations, integrators, and federal agencies on their application security initiatives, we understand what's involved in helping organizations improve their ability to produce and deploy secure applications cost effectively. Technical leaders and managers who are responsible for the security of a single application or a portfolio of applications will learn how to get organized, understand the threat, manage projects to generate assurance, and deliver meaningful metrics to senior management. We discuss techniques for creating a culture that encourages secure software, and how to support that culture with processes and technologies. Managing application security requires a balanced approach. We introduce the key activities required to create and maintain that balance by focusing on four key areas: foundation, implementation, verification, and management.

421 Application Security Risk Management *[14:58 minutes]*

Application Security Risk Management is the cornerstone of a successful application security program. This module covers the basic standards that an organization should have in place in order to organize their application portfolio, rate vulnerabilities, and schedule verification efforts. Application security risk management involves a series of activities across the organization and development lifecycle. We discuss the technical infrastructure you can use to manage risks effectively and generate useful metrics.

450 Integrating Security into Waterfall Projects *[22:18 minutes]*

Methods of integrating security activities into a waterfall-style project are explored. Activities such as security requirements, threat modeling, security architecture, secure coding, security testing, secure deployment, and secure operation are discussed. We suggest efficient approaches to generate the amount of assurance required. Students should have a working understanding of application security issues and software development processes.

460 Integrating Security into Agile Projects *[19:09 minutes]*

Learners are guided through the core foundations, activities, and job aides required to integrate security within Agile projects. The sprint structure of Agile projects makes them challenging for traditional approaches to application security, which track the stages in a waterfall-style project. Students will learn how to build security in a more Agile way and still achieve a sufficient level of assurance to allow an application to be approved for operation. Students should have a working understanding of application security issues and Agile software methods.

Pricing Summary

GSA Pricing Details
Contract GS-35F-082U
Modification Effective 03-16-2015

SIN 132-50 Instructor Lead Training:

10% Discount (Two day class = $20,000, GSA Price = $18,000 (if only one ordered)
Additional Quantity Discount, 3 or more classes ordered at a time 10%, 10 or more 20%, 50 or more 30%

SIN 132-51 Professional Services:

GSA FIXED HOURLY RATES:

			Prices have 2.0% annual escalation		
Job Title	First Year (2013)	Second Year (2014)	Third Year (2015)	Fourth Year (2016)	Fifth Year (2017)
Sr. Security and/or Software Engineer	$209.42	$213.60	$217.88	$222.23	$226.68
Security and/or Software Engineer	$154.21	$157.30	$160.44	$163.65	$166.93
Systems Administrator	$131.05	$133.67	$136.34	$139.07	$141.85
Assoc. Security and/or SW Engineer	$116.72	$119.05	$121.43	$123.86	$126.34

SIN 123-32 Term Software Licenses:

Contrast On-Premise	Applications 1 – 10	$9,700 per year per application
	Applications 11+	$4,850 per year per application

SIN 132-50 Application Security eLearning Modules:

Pricing is an annual subscription

	Volume Discount	0%	20%	35%	60%	
	# Modules	**99 or less Base per User**	**100-499 Cost/User**	**500 to 899 Cost/User**	**900+ Cost/User**	**Enterprise – No extra fee after 900 users**
APPLICATION SECURITY AWARENESS SERIES	7	$54.32	$43.46	$35.31	$21.73	$19,555.20
SECURE SOFTWARE DEVELOPMENT SERIES	35	$271.60	$217.28	$176.54	$108.64	$97,776.00
SECURE ARCHITECURES AND THREAT MODELING SERIES	6	$46.56	$37.25	$30.26	$18.62	$16,761.60
APPLICATION SECURITY TESTING SERIES	4	$31.04	$24.83	$20.18	$12.42	$11,174.40
LEADERS & MANAGERS SERIES	5	$38.80	$31.04	$25.22	$15.52	$13,968.00
Enterprise (all modules)	57	$354.05	$283.24	$230.13	$141.62	$127,458.00

Renewals are 100% of the annual fee